

```

class LinkedList:
    linked_list = []

    def __init__(self, *args):
        for arg in args:
            self.linked_list.append(int(arg))
        print("Linked List successfully created.\n")

    def display_linked_list(self):
        for data in self.linked_list:
            print(f"{data} ->", end=" ")
        print("NULL\n")

    def insert_node(self):
        if len(self.linked_list) != 0:
            k = int(input("\nEnter the index where you want to insert a node:
"))
            while(k < 0 or k > len(self.linked_list)):
                k = int(input(f>Please enter index between (0-
{len(self.linked_list)}): "))
            else:
                k = 0
            self.linked_list.insert(k, int(input("Enter your data for new node:
")))
            print("Node successfully inserted!\n")

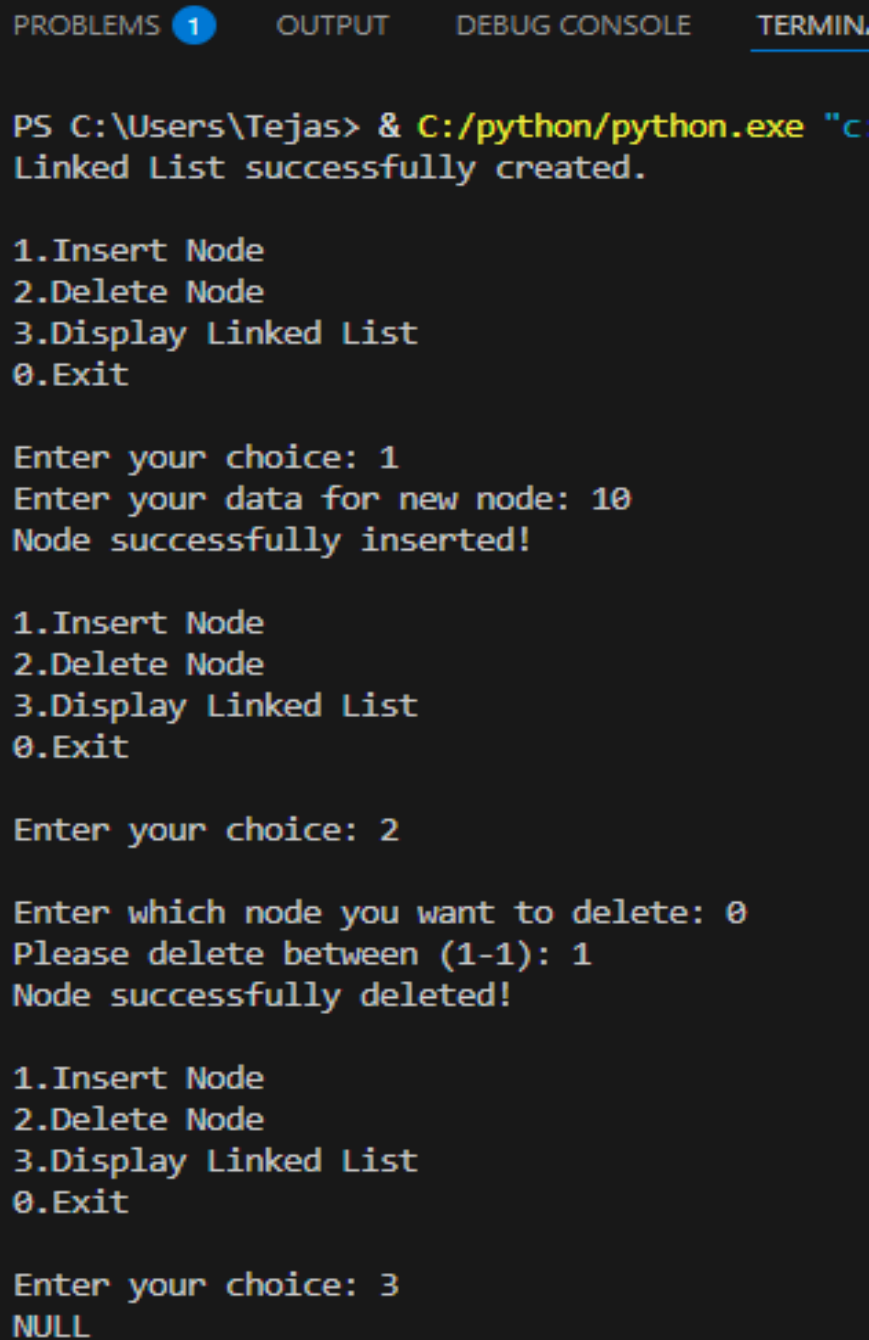
        def delete_node(self):
            if len(self.linked_list) != 0:
                k = int(input("\nEnter which node you want to delete: "))
                while(k < 1 or k > len(self.linked_list)):
                    k = int(input(f>Please delete between (1-
{len(self.linked_list)}): "))
                self.linked_list.pop(k-1)
                print("Node successfully deleted!\n")
            else:
                print("You cannot delete elements from an empty list\n")

user_linked_list = LinkedList()
choice = -1
while choice != 0:
    print("1.Insert Node\n2.Delete Node\n3.Display Linked List\n0.Exit\n")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        user_linked_list.insert_node()
    elif choice == 2:
        user_linked_list.delete_node()
    elif choice == 3:
        user_linked_list.display_linked_list()

```

```
elif choice == 0:  
    print("Exiting the program...")  
else:  
    print("Please enter a valid number\n")  
exit(0)
```

Output:



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL  
PS C:\Users\Tejas> & C:/python/python.exe "c  
Linked List successfully created.  
  
1.Insert Node  
2.Delete Node  
3.Display Linked List  
0.Exit  
  
Enter your choice: 1  
Enter your data for new node: 10  
Node successfully inserted!  
  
1.Insert Node  
2.Delete Node  
3.Display Linked List  
0.Exit  
  
Enter your choice: 2  
  
Enter which node you want to delete: 0  
Please delete between (1-1): 1  
Node successfully deleted!  
  
1.Insert Node  
2.Delete Node  
3.Display Linked List  
0.Exit  
  
Enter your choice: 3  
NULL
```

```

class Queue:
    queue = []
    def __init__(self, *args):
        for arg in args:
            self.queue.append(int(arg))
        print("Queue successfully created\n")

    def enqueue(self):
        self.queue.append(int(input("Enter you data: ")))
        print("\nData Enqueued!\n")

    def dequeue(self):
        if len(self.queue) != 0:
            self.queue.pop(0)
            print("\nData Dequeued!\n")
        else:
            print("\nUnderflow\n")

    def show(self):
        if len(self.queue) != 0:
            print("\nQueue elements (1st -> Last):", end=" ")
            for data in self.queue:
                print(f"{data}", end=" ")
            print("\n")
        else:
            print("\nYou cannot display a empty queue!\n")

user_queue = Queue()
choice = -1
while choice != 0:
    print("1.Enqueue\n2.Dequeue\n3.Show Queue\n0.Exit\n")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        user_queue.enqueue()
    elif choice == 2:
        user_queue.dequeue()
    elif choice == 3:
        user_queue.show()
    elif choice == 0:
        print("Exiting the program...")
    else:
        print("Please enter a valid number\n")
exit(0)

```

Output:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TESTS
Queue successfully created

1.Enqueue
2.Dequeue
3.Show Queue
0.Exit

Enter your choice: 1
Enter you data: 20

Data Enqueued!

1.Enqueue
2.Dequeue
3.Show Queue
0.Exit

Enter your choice: 1
Enter you data: 10

Data Enqueued!

1.Enqueue
2.Dequeue
3.Show Queue
0.Exit

Enter your choice: 3

Queue elements (1st -> Last): 20 10
```

```

class Bank:
    def __init__(self):
        print("\nBank Portal successfully opened!\n")

        customer_accounts = {}

    def create_account(self, username, password):
        if username not in self.customer_accounts.keys():
            self.customer_accounts[username] = [password, 0]
            print("\nAccount successfully created!\n")
        else:
            username = str(input("\nUsername already taken! Please try another username: "))
            self.create_account(username, password)

    def account_login(self, username, password):
        if len(self.customer_accounts) != 0:
            if username in self.customer_accounts.keys():
                if self.customer_accounts[username][0] == password:
                    print("\nAccount successfully opened!\n")
                    self.username = username
                    return -1
                else:
                    print("\nIncorrect password!\n")
                    return 0
            else:
                print("\nIncorrect username\n")
                return 0
        else:
            print("\nNo accounts created! Please create a account and try again!\n")
            return 0

    def deposit(self, deposit_amount):
        (self.customer_accounts[self.username])[1] += deposit_amount
        print("\nAmount successfully deposited! Transaction Completed!\n")

    def withdraw(self, withdraw_amount):
        if (self.customer_accounts[self.username])[1] >= withdraw_amount:
            (self.customer_accounts[self.username])[1] -= withdraw_amount
            print("\nAmount successfully withdrawn! Transaction Completed!\n")
        else:
            print(f"\nYou cannot withdraw more than your balance! Balance = {(self.customer_accounts[self.username])[1]}\n")

    def show_balance(self):
        print(f"\nYour account balance is ${{(self.customer_accounts[self.username])[1]:.3f}}\n")

```

```

my_bank = Bank()
choice = -1
while choice != 0:
    choice = int(input("1.Create New Account\n2.Account Login\n0.Exit\n\nEnter
your choice: "))

    if choice == 1:
        username = str(input("\nEnter your account username: "))
        password = str(input("Enter your account password: "))
        my_bank.create_account(username, password)

    elif choice == 2:
        username = str(input("\nEnter your account username: "))
        password = str(input("Enter your account password: "))
        inner_choice = my_bank.account_login(username, password)

        while inner_choice != 0:
            inner_choice = int(input("1.Deposit\n2.Withdraw\n3.Check
Balance\n0.Exit\n\nEnter your choice: "))

            if inner_choice == 1:
                deposit_amount = float(input("Enter your deposit amount: "))
                while deposit_amount < 0:
                    deposit_amount = float(input("Please enter a positive
value: "))
                my_bank.deposit(deposit_amount)

            elif inner_choice == 2:
                withdraw_amount = float(input("Enter your withdraw amount: "))
                while withdraw_amount < 0:
                    withdraw_amount = float(input("Please enter a positive
value: "))
                my_bank.withdraw(withdraw_amount)

            elif inner_choice == 3:
                my_bank.show_balance()

            elif inner_choice == 0:
                print("\nLogging you out..\nSuccessfully logged out\n")

            else:
                print("\nPlease enter a valid number!\n")

    elif choice == 0:
        print("\nExiting the program...")

    else:

```

```
        print("\nPlease enter a valid number!\n")
exit(0)
```

Output:

Bank Portal successfully opened!

1.Create New Account
2.Account Login
0.Exit

Enter your choice: 1

Enter your account username: Vivaan
Enter your account password: 1234

Account successfully created!

1.Create New Account
2.Account Login
0.Exit

Enter your choice: 2

Enter your account username: Vivaan
Enter your account password: 1234

Account successfully opened!

1.Deposit
2.Withdraw
3.Check Balance
0.Exit

Enter your choice: 1
Enter your deposit amount: 3000

Amount successfully deposited! Transaction Completed!

Amount successfully deposited! Transaction Completed!

- 1.Deposit
- 2.Withdraw
- 3.Check Balance
- 0.Exit

Enter your choice: 2

Enter your withdraw amount: 1000

Amount successfully withdrawn! Transaction Completed!

- 1.Deposit
- 2.Withdraw
- 3.Check Balance
- 0.Exit

Enter your choice: 0

Logging you out..

Successfully logged out

- 1.Create New Account
- 2.Account Login
- 0.Exit

Enter your choice: 0

Exiting the program...


```

class Employee:
    def __init__(self, name = "", salary = 0.0):
        self.name = name
        self.salary = salary

    employees = []

    def add_employee(self):
        employee = Employee(str(input("Enter employee name: ")),
float(input("Enter employee salary: ")))
        self.employees.append(employee)

    def __add__(self, other):
        return self.salary + other.salary

    def __sub__(self, other):
        return self.salary - other.salary

    def __gt__(self, other):
        return self.salary > other.salary

    def __lt__(self, other):
        return self.salary < other.salary

dummy_employee1 = Employee("Dummy1", 10)
dummy_employee2 = Employee("Dummy2", 20)
list = Employee()
choice = -1
while choice != 0:
    print("1.Add new employee\n2.Add or Subtract salary of 2
employees\n3.Compare salary of 2 employees\n0.Exit\n")
    choice = int(input("Enter your choice: "))

    if choice == 1:
        list.add_employee()
        print("\nEmployee successfully added\n")

    elif choice == 2:
        inner_choice = -1
        while inner_choice != 0:
            print("\n1.Add employees salary\n2.Subtract employees
salary\n0.Exit\n")
            inner_choice = int(input("Enter your choice: "))

            if inner_choice == 1:
                name1 = str(input("Enter name of employee 1: "))
                name2 = str(input("Enter name of employee 2: "))

```

```

        employee1 = next((emp for emp in list.employees if emp.name ==
name1), dummy_employee1)
        employee2 = next((emp for emp in list.employees if emp.name ==
name2), dummy_employee2)

        if employee1 != dummy_employee1 and employee2 !=
dummy_employee2:
            print(f"\nThe addition of employee salaries is {employee1
+ employee2}")
        else:
            employee1 = dummy_employee1
            employee2 = dummy_employee2
            print("\n1 or 2 employees not found!\nUsing dummy
employees with salary 10 and 20")
            print(f"\nThe addition of employee salaries is {employee1
+ employee2}")

    elif inner_choice == 2:
        name1 = str(input("Enter name of employee 1: "))
        name2 = str(input("Enter name of employee 2: "))

        employee1 = next((emp for emp in list.employees if emp.name ==
name1), dummy_employee1)
        employee2 = next((emp for emp in list.employees if emp.name ==
name2), dummy_employee2)

        if employee1 != dummy_employee1 and employee2 !=
dummy_employee2:
            print(f"\nThe subtraction of employee salaries (Employee
1 - Employee 2) is {employee1 - employee2}")
        else:
            employee1 = dummy_employee1
            employee2 = dummy_employee2
            print("\n1 or 2 employees not found!\nUsing dummy
employees with salary 10 and 20\n")
            print(f"\nThe subtraction of employee salaries (Employee
1 - Employee 2) is {employee1 - employee2}")

    elif inner_choice == 0:
        print("Exiting operator menu...\n")

    else:
        print("\nPlease enter a valid number!\n")

elif choice == 3:
    inner_choice = -1
    while inner_choice != 0:
        print("\n1.Compare greater than\n2.Compare less than\n0.Exit\n")

```

```

        inner_choice = int(input("Enter your choice: "))

        if inner_choice == 1:
            name1 = str(input("Enter name of employee 1: "))
            name2 = str(input("Enter name of employee 2: "))

            employee1 = next((emp for emp in list.employees if emp.name ==
name1), dummy_employee1)
            employee2 = next((emp for emp in list.employees if emp.name ==
name2), dummy_employee2)

            if employee1 != dummy_employee1 and employee2 !=
dummy_employee2:
                print(f"\nSalary of (Employee 1 > Employee 2):
{employee1>employee2}")
            else:
                employee1 = dummy_employee1
                employee2 = dummy_employee2
                print("\n1 or 2 employees not found!\nUsing dummy
employees with salary 10 and 20")
                print(f"\nSalary of (Employee 1 > Employee 2):
{employee1>employee2}")

        elif inner_choice == 2:
            name1 = str(input("Enter name of employee 1: "))
            name2 = str(input("Enter name of employee 2: "))

            employee1 = next((emp for emp in list.employees if emp.name ==
name1), dummy_employee1)
            employee2 = next((emp for emp in list.employees if emp.name ==
name2), dummy_employee2)

            if employee1 != dummy_employee1 and employee2 !=
dummy_employee2:
                print(f"\nSalary of (Employee 1 < Employee 2):
{employee1<employee2}")
            else:
                employee1 = dummy_employee1
                employee2 = dummy_employee2
                print("\n1 or 2 employees not found!\nUsing dummy
employees with salary 10 and 20")
                print(f"\nSalary of (Employee 1 < Employee 2):
{employee1<employee2}")

        elif inner_choice == 0:
            print("Exiting comparison menu...\n")

        else:

```

```

        print("\nPlease enter a valid number!\n")

    elif choice == 0:
        print("\nExiting the program...")

    else:
        print("\nPlease enter a valid number!\n")
exit(0)

```

Output:

```

PS C:\Users\Tejas> & C:/python/python.exe "c:/
1.Add new employee
2.Add or Subtract salary of 2 employees
3.Compare salary of 2 employees
0.Exit

Enter your choice: 2

1.Add employees salary
2.Subtract employees salary
0.Exit

Enter your choice: 1
Enter name of employee 1: jief
Enter name of employee 2: jnefo

1 or 2 employees not found!
Using dummy employees with salary 10 and 20

The addition of employee salaries is 30

1.Add employees salary
2.Subtract employees salary
0.Exit

Enter your choice: 0
Exiting operator menu...

1.Add new employee
2.Add or Subtract salary of 2 employees
3.Compare salary of 2 employees
0.Exit

Enter your choice: 3

```

Enter your choice: 3

- 1.Compare greater than
- 2.Compare less than
- 0.Exit

Enter your choice: 1

Enter name of employee 1: jns

Enter name of employee 2: ewouhf

1 or 2 employees not found!

Using dummy employees with salary 10 and 20

Salary of (Employee 1 > Employee 2): False

- 1.Compare greater than
- 2.Compare less than
- 0.Exit

Enter your choice: 0

Exiting comparison menu...

- 1.Add new employee
- 2.Add or Subtract salary of 2 employees
- 3.Compare salary of 2 employees
- 0.Exit

Enter your choice: 0

Exiting the program...

```

from abc import ABC, abstractmethod
class Shape(ABC):
    @abstractmethod
    def area(self):
        pass

    @abstractmethod
    def perimeter(self):
        pass

class Rectangle(Shape):
    def __init__(self, length = 0.0):
        self.length = length

    def area(self):
        return self.length**2

    def perimeter(self):
        return 4 * self.length

class Circle(Shape):
    def __init__(self, radius = 0.0):
        self.radius = radius

    def area(self):
        return PI * (self.radius**2)

    def perimeter(self):
        return 2 * PI * self.radius

rectangle = Rectangle()
circle = Circle()

choice = -1
while choice != 0:
    print("1.Rectangle\n2.Circle\n0.Exit\n")
    choice = int(input("Enter your choice: "))
    if choice == 1:
        inner_choice = -1
        while inner_choice != 0:
            print("\n1.Set length\n2.Print Perimeter\n3.Print Area\n0.Exit\n")
            inner_choice = int(input("Enter your choice: "))

            if inner_choice == 1:
                rectangle.length = float(input("\nEnter the length of
rectangle: "))
                while rectangle.length < 0:

```

```

        rectangle.length = float(input("Please enter a positive
value: "))
        print("\nValue successfully updated!")

        elif inner_choice == 2:
            print(f"\nThe perimeter of rectangle is
{rectangle.perimeter():.3f}")

        elif inner_choice == 3:
            print(f"\nThe area of rectangle is {rectangle.area():.3f}")

        elif inner_choice == 0:
            print("\nExiting rectangle...\n")

        else:
            print("\nPlease enter a valid number!")

    elif choice == 2:
        inner_choice = -1
        while inner_choice != 0:
            print("\n1.Set Radius\n2.Print Circumference(Perimeter)\n3.Print
Area\n0.Exit\n")
            inner_choice = int(input("Enter your choice: "))

            if inner_choice == 1:
                circle.radius = float(input("\nEnter the radius of circle: "))
                while circle.radius < 0:
                    circle.radius = float(input("Please enter a positive
value: "))
                print("\nValue successfully updated!")

                elif inner_choice == 2:
                    print(f"\nThe perimeter of circle is
{circle.perimeter():.3f}")

                elif inner_choice == 3:
                    print(f"\nThe area of circle is {circle.area():.3f}")

                elif inner_choice == 0:
                    print("\nExiting circle...\n")

                else:
                    print("\nPlease enter a valid number!")

    elif choice == 0:
        print("Exiting the program...")

    else:

```

```
        print("\nPlease enter a valid number!\n")
    exit(0)
```

Output:

```
1.Rectangle
2.Circle
0.Exit

Enter your choice: 1

1.Set length
2.Print Perimeter
3.Print Area
0.Exit

Enter your choice: 1

Enter the length of rectangle: 20

Value successfully updated!

1.Set length
2.Print Perimeter
3.Print Area
0.Exit

Enter your choice: 2

The perimeter of rectangle is 80.000

1.Set length
2.Print Perimeter
3.Print Area
0.Exit

Enter your choice: 3

The area of rectangle is 400.000
```


- 1.Rectangle
- 2.Circle
- 0.Exit

Enter your choice: 2

- 1.Set Radius
- 2.Print Circumference(Perimeter)
- 3.Print Area
- 0.Exit

Enter your choice: 1

Enter the radius of circle: 5

Value successfully updated!

- 1.Set Radius
- 2.Print Circumference(Perimeter)
- 3.Print Area
- 0.Exit

Enter your choice: 2

The perimeter of circle is 31.416

- 1.Set Radius
- 2.Print Circumference(Perimeter)
- 3.Print Area
- 0.Exit

Enter your choice: 3

The area of circle is 78.540

```

class Bank:
    def __init__(self):
        print("\nBank Portal successfully opened!\n")
        self.customer_accounts = {}

    def create_account(self, account_number, password):
        if account_number not in self.customer_accounts.keys():
            self.customer_accounts[account_number] = [password, 0]
            print("\nAccount successfully created!\n")
        else:
            account_number = int(input("\nAccount number already exist! Please
try another number: "))
            self.create_account(account_number, password)

    def account_login(self, account_number, password):
        if len(self.customer_accounts) != 0:
            if account_number in self.customer_accounts.keys():
                if self.customer_accounts[account_number][0] == password:
                    print("\nAccount successfully opened!\n")
                    self.account_number = account_number
                    return -1
                else:
                    print("\nIncorrect password!\n")
                    return 0
            else:
                print("\nIncorrect account number\n")
                return 0
        else:
            print("\nNo accounts created! Please create a account and try
again!\n")
            return 0

    def deposit(self, deposit_amount):
        (self.customer_accounts[self.account_number])[1] += deposit_amount
        print("\nAmount successfully deposited! Transaction Completed!\n")

    def withdraw(self, withdraw_amount):
        if (self.customer_accounts[self.account_number])[1] >=
withdraw_amount:
            (self.customer_accounts[self.account_number])[1] -=
withdraw_amount
            print("\nAmount successfully withdrawn! Transaction Completed!\n")
        else:
            print(f"\nYou cannot withdraw more than your balance! Balance =
{(self.customer_accounts[self.account_number])[1]}\n")

    def show_details(self):
        print(f"\nYour account number: {self.account_number}")

```

```

        print(f"Balance:
${(self.customer_accounts[self.account_number])[1]:.3f}\n")

my_bank = Bank()
choice = -1
while choice != 0:
    choice = int(input("1.Create New Account\n2.Account Login\n0.Exit\n\nEnter
your choice: "))

    if choice == 1:
        account_number = int(input("\nEnter a account number: "))
        password = str(input("Enter your account password: "))
        my_bank.create_account(account_number, password)

    elif choice == 2:
        account_number = int(input("\nEnter your account number: "))
        password = str(input("Enter your account password: "))
        inner_choice = my_bank.account_login(account_number, password)

        while inner_choice != 0:
            inner_choice = int(input("1.Deposit\n2.Withdraw\n3.Check account
details\n0.Exit\n\nEnter your choice: "))

            if inner_choice == 1:
                deposit_amount = float(input("Enter your deposit amount: "))
                while deposit_amount < 0:
                    deposit_amount = float(input("Please enter a positive
value: "))
                my_bank.deposit(deposit_amount)

            elif inner_choice == 2:
                withdraw_amount = float(input("Enter your withdraw amount: "))
                while withdraw_amount < 0:
                    withdraw_amount = float(input("Please enter a positive
value: "))
                my_bank.withdraw(withdraw_amount)

            elif inner_choice == 3:
                my_bank.show_details()

            elif inner_choice == 0:
                print("\nLogging you out..\nSuccessfully logged out\n")

            else:
                print("\nPlease enter a valid number!\n")

    elif choice == 0:
        print("\nExiting the program...")

```

```
    else:  
        print("\nPlease enter a valid number!\n")  
exit(0)
```

Output:

```
Bank Portal successfully opened!  
  
1.Create New Account  
2.Account Login  
0.Exit  
  
Enter your choice: 1  
  
Enter a account number: 0987  
Enter your account password: 1234  
  
Account successfully created!  
  
1.Create New Account  
2.Account Login  
0.Exit  
  
Enter your choice: 2  
  
Enter your account number: 0987  
Enter your account password: 1234  
  
Account successfully opened!  
  
1.Deposit  
2.Withdraw  
3.Check account details  
0.Exit  
  
Enter your choice: 1  
Enter your deposit amount: 90000  
  
Amount successfully deposited! Transaction Completed!
```

1.Deposit
2.Withdraw
3.Check account details
0.Exit

Enter your choice: 2

Enter your withdraw amount: 30000

Amount successfully withdrawn! Transaction Completed!

1.Deposit
2.Withdraw
3.Check account details
0.Exit

Enter your choice: 3

Your account number: 987

Balance: \$60000.000

1.Deposit
2.Withdraw
3.Check account details
0.Exit

Enter your choice: 0

Logging you out..

Successfully logged out

1.Create New Account
2.Account Login
0.Exit

Enter your choice: 0

```

import math

class Vector2D:
    def __init__(self, x = 0, y = 0, z = 0):
        self.vector = [x, y, z]

    def magnitude(self):
        magnitude = 0.0
        for i in range(3):
            magnitude += self.vector[i]**2
        magnitude **= 0.5
        return magnitude

    def distance(self, other_vector):
        distance = 0.0
        for i in range(3):
            distance += (self.vector[i] - other_vector[i])**2
        distance **= 0.5
        return distance

    def angle_x(self):
        angle_x = math.acos(self.vector[0] / self.magnitude()) * 180 / 3.14159
        return angle_x

    def angle_y(self):
        angle_y = math.acos(self.vector[1] / self.magnitude()) * 180 / 3.14159
        return angle_y

    def angle_z(self):
        angle_z = math.acos(self.vector[2] / self.magnitude()) * 180 / 3.14159
        return angle_z

    def dot_product(self, other_vector):
        dot_product = 0.0
        for i in range(3):
            dot_product += self.vector[i] * other_vector[i]
        return dot_product

    def cross_product(self, other_vector):
        z = self.vector[0] * other_vector[1] - self.vector[1] *
other_vector[0]
        return f"{z}k"

class Vector3D(Vector2D):
    def cross_product(self, other_vector):
        x = self.vector[1] * other_vector[2] - self.vector[2] *
other_vector[1]

```

```

        y = self.vector[2] * other_vector[0] - self.vector[0] *
other_vector[2]
        z = self.vector[0] * other_vector[1] - self.vector[1] *
other_vector[0]
        return f"({x})i + ({y})j + ({z})k"

vector2d_1 = Vector2D()
vector2d_2 = Vector2D()
vector3d_1 = Vector3D()
vector3d_2 = Vector3D()

choice = -1
while choice != 0:
    print("1. 2-D Vector\n2. 3-D Vector\n0. Exit\n")
    choice = int(input("Enter your choice: "))

    if choice == 1:
        inner_choice = -1
        while inner_choice != 0:
            print("1.Update Vector 1 (Main Vector)\n2.Update Vector
2\n3.Calculate Magnitude\n4.Calculate Angle\n5.Calculate Distance")
            print("6.Calculate Dot Product\n7.Calculate Cross
Product\n0.Exit\n")
            inner_choice = int(input("Enter your choice: "))

            if inner_choice == 1:
                x,y = input("Enter your x and y coordinates: ").split()
                x = int(x)
                y = int(y)
                vector2d_1 = Vector2D(x, y)

            elif inner_choice == 2:
                x, y = input("Enter your x and y coordinates: ").split()
                x = int(x)
                y = int(y)
                vector2d_2 = Vector2D(x, y)

            elif inner_choice == 3:
                print(f"Magnitude of Vector 1:
{vector2d_1.magnitude():.3f}\n")

            elif inner_choice == 4:
                print(f"Angle with x-axis: {vector2d_1.angle_x():.3f}")
                print(f"Angle with y-axis: {vector2d_1.angle_y():.3f}\n")

            elif inner_choice == 5:
                print(f"Distance between vectors:
{vector2d_1.distance(vector2d_2.vector):.3f}\n")

```

```

        elif inner_choice == 6:
            print(f"Dot Product:
{vector2d_1.dot_product(vector2d_2.vector)}\n")

        elif inner_choice == 7:
            print(f"Cross Product:
{vector2d_1.cross_product(vector2d_2.vector)}\n")

        elif inner_choice == 0:
            print("Exiting the menu\n")
            break

    elif choice == 2:
        inner_choice = -1
        while inner_choice != 0:
            print("1.Update Vector 1 (Main Vector)\n2.Update Vector
2\n3.Calculate Magnitude\n4.Calculate Angles")
            print("5.Calculate Distance\n6.Calculate Dot Product\n7.Calculate
Cross Product\n0.Exit\n")
            inner_choice = int(input("Enter your choice: "))

        if inner_choice == 1:
            x, y, z = input("Enter your x, y, z coordinates: ").split()
            x = int(x)
            y = int(y)
            z = int(z)
            vector3d_1 = Vector3D(x, y, z)

        elif inner_choice == 2:
            x, y, z = input("Enter your x, y, and z coordinates:
").split()
            x = int(x)
            y = int(y)
            z = int(z)
            vector3d_2 = Vector3D(x, y, z)

        elif inner_choice == 3:
            print(f"Magnitude of Vector 1: {vector3d_1.magnitude():.3f}")

        elif inner_choice == 4:
            print(f"Angle with x-axis: {vector3d_1.angle_x():.3f}")
            print(f"Angle with y-axis: {vector3d_1.angle_y():.3f}")
            print(f"Angle with z-axis: {vector3d_1.angle_z():.3f}")

        elif inner_choice == 5:
            print(f"Distance between vectors:
{vector3d_1.distance(vector3d_2.vector):.3f}")

```



```

        elif inner_choice == 6:
            print(f"Dot Product:
{vector3d_1.dot_product(vector3d_2.vector)}")

        elif inner_choice == 7:
            print(f"Cross Product:
{vector3d_1.cross_product(vector3d_2.vector)}")

        elif inner_choice == 0:
            print("Exiting the menu\n")
            break

    elif choice == 0:
        print("\nExiting the program...")
        break

    else:
        print("\nPlease enter a valid number!\n")

```

Output:

```

1. 2-D Vector
2. 3-D Vector
0. Exit

Enter your choice: 2
1.Update Vector 1 (Main Vector)
2.Update Vector 2
3.Calculate Magnitude
4.Calculate Angles
5.Calculate Distance
6.Calculate Dot Product
7.Calculate Cross Product
0.Exit

Enter your choice: 1
Enter your x, y, z coordinates: 1 2 3
1.Update Vector 1 (Main Vector)
2.Update Vector 2
3.Calculate Magnitude
4.Calculate Angles
5.Calculate Distance
6.Calculate Dot Product
7.Calculate Cross Product
0.Exit

Enter your choice: 2
Enter your x, y, and z coordinates: 5 2 8
1.Update Vector 1 (Main Vector)
2.Update Vector 2
3.Calculate Magnitude
4.Calculate Angles
5.Calculate Distance
6.Calculate Dot Product
7.Calculate Cross Product
0.Exit

```

```
Enter your choice: 7
Cross Product: (10)i + (7)j + (-8)k
1.Update Vector 1 (Main Vector)
2.Update Vector 2
3.Calculate Magnitude
4.Calculate Angles
5.Calculate Distance
6.Calculate Dot Product
7.Calculate Cross Product
0.Exit
```

```
Enter your choice: 5
Distance between vectors: 6.403
1.Update Vector 1 (Main Vector)
2.Update Vector 2
3.Calculate Magnitude
4.Calculate Angles
5.Calculate Distance
6.Calculate Dot Product
7.Calculate Cross Product
0.Exit
```

```
Enter your choice: 0
Exiting the menu
```

```
1. 2-D Vector
2. 3-D Vector
0. Exit
```

```
Enter your choice: 0
Exiting the program...
```

```

def decode(s):
    def helper(s, memo):
        if s in memo:
            return memo[s]
        if not s:
            return [""]
        res = []
        if s[0] != "0":
            res += [chr(96 + int(s[:1])) + item for item in helper(s[1:],
memo)]
            if len(s) >= 2 and "10" <= s[:2] <= "26":
                res += [chr(96 + int(s[:2])) + item for item in helper(s[2:],
memo)]
        memo[s] = res
        return res

    return helper(s, {})

encoded_message = str(input("Enter your code: "))
decoded_messages = decode(encoded_message)
print(decoded_messages)

```

Output:

```

PS C:\Users\Tejas> & C:/python
Enter your code: 11106
['aajf', 'kjf']
PS C:\Users\Tejas>

```

```
import re
```

```

class tokenizer_hindi:
    def __init__(self, text):
        self.t = text
    def punctuations(self):
        '''tokenize punctuations'''
        pattern = r'[|,;?!""-:^°=...]'
        res = re.split(pattern, self.t)
        return res
    def dates(self):
        '''dates from text'''
        pattern = r'[\d]+/[\d]+/[\d]+'
        return re.findall(pattern, self.t)
    def urls(self):
        '''returns url from text'''
        pattern = r'http[s]*://www.[\w]+.com'
        return re.findall(pattern, self.t)
    def emails(self):
        '''returns emails from text'''
        pattern = r'[\w]+@[\w]+.com'
        return re.findall(pattern, self.t)
    def numbers(self):
        '''returns various numbers from text'''
        pattern = r'[\d]+/[\d]+|[\d]+.[\d]+|\s[\d]+\s'
        return re.findall(pattern, self.t)
    def username(self):
        '''returns social media usernames handle names'''
        pattern = r'@[\w]+'
        return re.findall(pattern, self.t)
    def display(self):
        return self.t
    def whole(self):
        pattern = r'[|,;?!""-:^°=...]|[\d]+/[\d]+/[\d]+|http[s]*://www.[\w]+.com|[\w]+@[\w]+.com|[\d]+/[\d]+|[\d]+.[\d]+|\s[\d]+\s|@[\w]+|[\w]+'
        return re.findall(pattern, self.t)

a = tokenizer_hindi(''''नमस्ते! यह एक उदाहरण पाठ है। यहाँ हम विभिन्न विराम चिन्हों का
उपयोग करेंगे, जैसे अल्प विराम (,), पूर्ण विराम (।), और प्रश्नवाचक चिन्ह (?).
आज की तारीख 25/02/2025 है। क्या आप इसे पढ़ सकते हैं?
आप मुझे example@example.com पर संपर्क कर सकते हैं। यहाँ एक वेबसाइट भी है:
https://www.example.com
ट्विटर पर मेरा उपयोगकर्ता नाम है @example_user।
हमारे पास 3 किलो सेब, 7 संतरे, और 2.5 किलो अंगूर हैं।''')

print(a.punctuations())
print(a.dates())
print(a.urls())
print(a.emails())

```

```
print(a.numbers())
print(a.username())
```

Output:

```
PS C:\Users\Tejas> & C:/python/python.exe "c:/Vivaan/Programming/Python/Lab 8/Assignment8,9.py"
['नमस्ते', ' यह एक दाहरा पाठ है यहाँ हम वभिन्न वीराम ॥ न्हा ॥ प्रयोग करेंगे, ' जैसे लप वीराम', ',', ',', ' पूरा वीराम', '।', ',', ' पर परसव ॥ ॥ न्ह', ',', ',', '\n ॥ ॥ तार ॥', ',', ',', ',', ',', ',', ',', ',', ',', ' है। क्या ॥ प इसे पढ़ ॥ ते है', '\n ॥ प मुझे example@example', 'com पर संस् ॥ र ॥ ते है। यहाँ एक वेब।इट भी है, ' https', ',', ',', 'www', 'example', 'com\n त्वं। र पर मे। ॥ प्रयोग। र्ता नाम है @example_user\n हमारे पाठ', ' ॥ ॥ सेव', ',', ',', ' संस्', '। र', ',', ',', ' ॥ ॥ और है।']
['25/02/2025']
['https://www.example.com']
['example@example.com']
['3', '7', '2.5']
['@example', '@example_user']
PS C:\Users\Tejas>
```